

FREQUENCY LOOK-AHEAD AND LINK STATE HISTORY BASED SCHEDULING IN INDOOR WIRELESS PICO-CELLULAR NETWORKS

DESCRIPTION

5 BACKGROUND OF THE INVENTION

Field of the Invention

09452802-120299
662021-2085450

The present invention generally relates to time division duplex indoor wireless communication networks and, more particularly, to a frequency look-ahead, packet/slave scheduling scheme and master/slave link characterization using a link state history table in the master unit in order to account for channel and system characteristics.

10

Background Description

Bluetooth™ is a computing and telecommunications industry specification that describes how mobile phones, computers, personal digital assistants (PDAs), and other devices can interconnect using a short range wireless connection. Each device is equipped with a microchip transceiver that transmits and receives in the frequency band of 2.45 GHz. Each device will have a unique 48-bit address from the IEEE 802 standard. Connections are one-to-one, and the maximum range is ten meters. Data can be exchanged at a rate of one megabits per second (Mbs) and up to two Mbs in the second generation of the technology. The five founding companies of the Bluetooth™

15

20

Special Interest Group (SIG) are Ericsson, IBM, Intel, Nokia, and Toshiba. Additional information may be had by reference to the Web site www.bluetooth.com and an article by Andrew Seybold entitled "Bluetooth Technology: The Convergence of Communications and Computing", reprinted from *Andrew Seybold's Outlook*, May 1998, on the World Wide Web at www.gsmdata.com/artblue.htm.

Indoor wireless networks based on standards such as Bluetooth™ use frequency hopping to combat the problem of interference from sources such as microwave ovens and cordless telephones, which also use frequencies in the same band. In practical environments, in addition to active interfering sources, there can also be objects such as water fountains and racks of bottles with water content which absorb much of the radiation in the 2.45 GHz band and obstruct communication between master and slave units in the vicinity. Therefore, a master unit needs to detect such problems in communication and take necessary actions to prevent loss of packets during the periods of interference.

In the current Bluetooth™ standard, due to frequency hopping, the carrier frequency used in consecutive time slots is a different one of several different frequencies within the 2.45 GHz band of frequencies. Therefore, an interference in sub-bands centered around one of these frequencies will only affect communication during that time-slot in which the frequency sub-band is used. Further, in the Bluetooth™ standard, a packet can occupy one, three or five time slots, and in the case of multiple size packets, the same frequency as fixed for the first time slot is used. Because of this, it is possible to mask the effect of an interference by transmitting a packet of appropriate size. For example, if it is known that there is high chance of interference in one of the second through fifth time slots, and very low probability of the first time slot being bad, it is possible to skip the frequencies corresponding to second

through fifth time slots by transmitting a five time-slot packet instead of one or three time-slot packets.

The characterization of link between any slave unit and the master can be done by the master unit based on the receipt or otherwise of
5 acknowledgments received from the slave. Alternatively, all the slaves can record the number of times they detect good packet headers sent by the master to any slave. This information can be transmitted from the slaves to the master at periodic intervals of time. The master can use this information along with frequency look-ahead to determine the next slave for communication and also
10 the appropriate packet size.

Several methods and schemes to combat the effect of interference in cellular wireless communication systems have been proposed. For example, a Fast Fourier Transform (FFT) based adaptive interference cancellation method has been proposed in U.S. Patent Number 5,612,978 to Blanchard et al for
15 "Method and Apparatus for Real-time Adaptive Interference Cancellation in Dynamic Environments". The method allows relatively fast changes in the interference environment to be tracked and rejected.

In the U.S. Patent Number 5,541,954 to Emi et al. for "Frequency Hopping Communication Method and Apparatus Changing a Hopping
20 Frequency as a Result of a Counted Number of Errors", a technique of changing the hopping frequency based on the number of errors encountered on a given frequency is proposed. This scheme is suitable in systems where the hopping frequency can be changed at any time.

Another related invention can be found in U.S. Patent Number
25 5,323,447 to Gillis et al. for "Apparatus and Method for Modifying a Frequency Hopping Sequence of a Cordless Telephone Operating in a Frequency Hopping Domain", in which the hopping sequence is modified for a cordless telephone. Substitute alternative communication channels are

identified and then substituted for those communication channels experiencing interference. This is done without disrupting the communications between the handset unit and its associated base unit.

5 In systems in which the hopping frequency is fixed depending on the frequency hopping sequence corresponding any given pico-cell, there is need for other methods to combat the problem of interference. In U.S. Patent Number 5,570,352 to Pöyhönen for "Digital Cellular Network/System with Mobile Stations Communicating with Base Stations Using Frequency-Hopping and Having Enhanced Effect of Interference Diversity", frequency
10 hopping utilizing interference diversity is presented.

In U.S. Patent Number 5,659,879 to Dupuy for "Method of Covering Shadow Areas in a Cellular Mobile Radio and Radio Booster for Implementing this Method", the problem of covering shadow areas by using radio boosters is addressed. To cover the shadow areas, a radio signal received
15 by a radio booster from the base transceiver station on a basic frequency is re-transmitted to a mobile station on a translated frequency different from the basic frequency and associated with the latter by a translation law.

SUMMARY OF THE INVENTION

20 It is therefore an object of the present invention to provide a method to combat the effect of interfering sources in frequency hopping based indoor wireless networks.

According to the invention, a method of combating the problem of interference from external sources and shadowing objects in indoor pico-cellular wireless networks utilizes frequency look-ahead and short-term
25 history about channel state with reference to different mobile units within a pico-cell. The method monitors the states of master-slave wireless

communication links through values recorded in link counters. Based on the recorded values, an appropriate slave is scheduled and the suitable packet size chosen to overcome the effect of interfering sources if any in the pico-cell.

BRIEF DESCRIPTION OF THE DRAWINGS

5 The foregoing and other objects, aspects and advantages will be better understood from the following detailed description of preferred embodiments of the invention with reference to the drawings, in which:

Figure 1 is a timing diagram showing frequency hopping time slots and frequencies;

10 Figure 2 is a timing diagram showing examples of multi-slot master/slave transmissions;

Figure 3 is a block diagram of a frequency selection unit (FSU);

Figure 4 is a block diagram of the interface between a processor and an additional frequency selection unit (FSU);

15 Figure 5 is a flowchart for sequential look-ahead for "k" time slot frequencies;

Figure 6 is a flowchart for updating the link state history counters;

Figures 7A, 7B and 7C, taken together, are a flowchart of slave selection and packet size determination using link state history counters;

20 Figures 8A and 8B, taken together, are a flowchart of slave selection and packet size determination using goodness counter values;

Figure 9 is a timing diagram showing transmission time within a time slot; and

25 Figure 10 is a timing diagram showing a master's next transmission time slots.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS OF THE INVENTION

In the current Bluetooth™ indoor wireless networking standard, the master unit and slave units in any given pico-cell use the same frequency hopping sequence to communicate with each other. In each unit, there is a frequency selection unit (FSU) which produces a frequency index for one of several possible frequencies. The inputs to the frequency selection unit are the address bits of the pico-cell master and the current clock bits in each unit. When a master or slave has to transmit or receive information over the wireless medium, the frequency index is used to synthesize the appropriate frequency in the frequency hopping sequence of the pico-cell. As all the units in a pico-cell use the same address of the master for determining the frequency in any slot, the same hopping sequence is used by all of them. However, as the clocks of the different units are not synchronized, each unit uses a corresponding offset with reference to the clock of the master.

Each time slot in the Bluetooth™ standard occupies 625 micro-seconds. The master always begins transmission on an even time slot, and a slave always begins transmission on an odd time slot. A slave unit can transmit to the master on an up-link slot only if the master had transmitted to that slave on the preceding down-link slot. The frequencies used on adjacent time slots by master or slave units are chosen to be different based on the frequency hopping sequence for the pico-cell.

A schematic representation of hopping frequencies used during consecutive time slots is shown in Figure 1. Here, for simplicity in illustration only one-slot packets have been shown. However, in general, a master or slave can also transmit packets of size three or five time-slots. Also, the frequency used during the entire three or five size packet transmission is the same as that

for the first time slot in which the packet transmission started, as seen in the examples for frequencies used during multi-slot packet transmissions shown in Figure 2. It is to be noted that in the currently proposed standard, there is no frequency look-ahead with reference to the frequency hopping sequence.

5

Frequency Look-Ahead

A preferred embodiment of the invention comprises frequency look-ahead to know at any point of time the frequencies that will be used during the future time slots by the master and slave units. A frequency selection unit (FSU) as shown in block diagram form in Figure 3 is used to find the current
10 frequency on which to transmit or receive. The FSU receives address bits and clock bits at slot " i " and outputs an index to the frequency at slot " i ". The FSU is basically a combinatorial logic unit comprising functions of addition, exclusive-OR, and permutation operations.

In the present invention, an additional FSU is used to find the
15 frequencies corresponding to future time slots, as shown in block diagram form in Figure 4. The processor 41 in the master unit obtains the frequency corresponding to any time slot by providing as inputs to the additional FSU 42 the address bits and the appropriate clock bits. With this scheme, frequencies corresponding to next " k " time slots are obtained by the processor
20 sequentially.

The process is shown in the flowchart shown in Figure 5. The process begins by setting k to be equal to the number of look-ahead slots in function block 51 and setting i to be equal to the first look-ahead slot in function block 52. After this initialization, the process enters a processing loop which begins
25 with the processor (41 in Figure 4) outputting clock bits at slot " i " to the FSU (42 in Figure 4) in function block 53. Next, the processor gets the index to the

frequency at slot “ i ” from the processor in function block 54. “ i ” is set to the next look-ahead slot in function block 55. A test is made in decision block 56 to determine if all the required “ k ” indices have been obtained. If not, the process loops back to function block 53; otherwise, the process stops.

- 5 While look-ahead for sequential time slots is implied in this flowchart, the master can in general look-ahead for future time slots in any arbitrary order by just providing the clock bits corresponding to the required time slots.

Link State History

- 10 Another feature of the invention is maintaining the information about link state with reference to each frequency of operation and each active slave. In the present invention, this information is contained in a counter corresponding to each slave-frequency pair. The master unit maintains a table of counters as shown in Table 1. In Table 1, “ $C(i,j)$ ” represents the counter for link state of frequency f_j for slave i .

15 Table 1. Link State History Table of Counters

	f_1	f_2	f_3		f_{78}	f_{79}
SLAVE 1	C(1,1)	C(1,2)	C(1,3)	...	C(1,78)	C(1,79)
SLAVE 2	C(2,1)	C(2,2)	C(2,3)	...	C(2,78)	C(2,79)
SLAVE 3	C(3,1)	C(3,2)	C(3,3)	...	C(3,78)	C(3,79)
SLAVE 4	C(4,1)	C(4,2)	C(4,3)	...	C(4,78)	C(4,79)
20 SLAVE 5	C(5,1)	C(5,2)	C(5,3)	...	C(5,78)	C(5,79)
SLAVE 6	C(6,1)	C(6,2)	C(6,3)	...	C(6,78)	C(6,79)
SLAVE 7	C(7,1)	C(7,2)	C(7,3)	...	C(7,78)	C(7,79)

Here, seven rows corresponding to seven slaves have been shown because, in each pico-cell a maximum of seven active slaves are allowed at any time. However, the number of rows in the table during any period of time is equal to the number of active slaves in the pico-cell during that period.

5 Also, the number of columns in the table is equal to the number of hopping frequencies used in the system. In the current standard, the number of hopping frequencies is 79 (or 23 when operating in some countries of the world).

Initially, all the counters are reset to zero, and during transmissions the counter values are changed depending on the success or failure of a packet transmission between the master and a slave. Counter $C(i,j)$ and $C(i,k)$ corresponding to slave “ i ” and frequencies f_j and f_k are updated as shown in the flowchart of Figure 6. Here, f_j is the frequency of transmission by the master and f_k is the frequency on which a slave’s packet is received. In decision block 61, a test is made to determine if a packet has been received from slave “ i ”. If not, the counter $C(i,j)$ is incremented by one in function block 62. If the value of the counter exceeds a threshold value T_{RESET} after being incremented, the counter is reset. If the packet has been received, then a further test is made in decision block 63 to determine if the packet is corrupted. If so, the counters $C(i,j)$ and $C(i,k)$ are each incremented by one in function block 64. If the values of either of the counters exceeds the threshold value T_{RESET} after being incremented, the counter is reset. If the received packet is not corrupted, a further test is made in decision block 65 to determine if the master received an acknowledgment. If not, the counter $C(i,j)$ is incremented by one and the counter $C(i,k)$ is reset in function block 66. If the value of the counter $C(i,j)$ exceeds a threshold value T_{RESET} after being incremented, it is reset. If the master received an acknowledgment, the counters $C(i,j)$ and $C(i,k)$ are reset to zero in function block 67.

As shown in Figure 6, both counters $C(i,j)$ and $C(i,k)$ are incremented by one when either no packet or a corrupted packet is received. Also, counter $C(i,j)$ is incremented by one when the master receives a no-acknowledgment signal through a correctly received packet. Also as indicated in the flowchart,

5 counter $C(i,j)$ or counter $C(i,k)$ are reset to zero if their count value is greater than a threshold T_{RESET} after an increment operation. When an acknowledgment signal is received through a correctly received slave's packet, it indicates that the master's transmission was successfully received and the slave's transmission frequency also has no interference. Therefore,

10 both the counters $C(i,j)$ and $C(i,k)$ are reset to zero.

At the time of deciding about transmission to a slave "i", if counter $C(i,j)$ value is less than or equal to a threshold value T_{TRANSMIT} , a packet transmission is made to the slave. However, if counter $C(i,j)$ value is greater than T_{TRANSMIT} , then no transmission is sent to slave "i" at frequency f_j during

15 the scheduled time slot for the slave, and the counter $C(i,j)$ is incremented by one. After the increment operation, if value of counter $C(i,j)$ is greater than a threshold T_{RESET} , the counter is reset to zero. The counters $C(i,j)$ start with an initial value of zero, and with progressive communications between the master and the slaves, assume different values depending on the relative positions of

20 the mobile units and the nature and location of interfering sources. The two counter thresholds T_{TRANSMIT} and T_{RESET} are useful to distinguish between transient and persistent errors on a link. If a master-slave link on a certain frequency is affected with very short-term interference, then before the threshold T_{TRANSMIT} is exceeded, the master gets an acknowledgment for one of

25 its packets sent to the slave, and the corresponding counter is reset. However, if the inference is persistent, with repeated transmission failures, the threshold T_{TRANSMIT} is soon reached and no further attempts to transmit to the slave at that frequency are made for $(T_{\text{RESET}} - T_{\text{TRANSMIT}})$ scheduled time slots for the

slave. Thus, for each master-slave link T_{TRANSMIT} consecutive attempts are allowed and thereafter $(T_{\text{RESET}} - T_{\text{TRANSMIT}})$ period of back-off time is provided. During this back-off period, other slaves can be scheduled for transmission. The values of T_{TRANSMIT} and T_{RESET} are fixed depending on the desired number of transmission attempts at any frequency and the required back-off time.

Slave Scheduling and Packet Size Determination

Another feature of the invention comprises the determination of the next slave for transmission and the packet size to be used. The procedures for this are embodied in flowcharts shown in Figures 7A to 7C. As illustrated in Figures 7A to 7C, before transmission to a slave, the master unit obtains the frequencies corresponding to the time slots which need to be used in the immediate future by using the look-ahead feature described above. As the frequency of transmission for the entire packet is the same as that for the first time slot even if a three or five size packet is transmitted, the master need only to check if the frequency corresponding to the first time slot is good for transmission. Further, as the slave to which the master has transmitted is required to transmit at the time slot immediately following the last time slot used by the master, that time slot should also be good if the communication between the master and the slave is to be successful. Therefore, frequency look-ahead is required for the first time slot after the last time slot used by the currently transmitting slave, and for second, fourth or sixth time slots if the master is to transmit one, three or five sized packets, respectively.

Slave Selection

The first processing loop shown in Figure 7A determines slave selection using link state history counters. The process begins with decision

block 701 in which a determination is made as to whether all slaves have been checked. If not, the next slave is picked as the current slave "i" in function block 702. A test is then made in decision block 703 as to whether the value of counter $C(i,j)$ exceeds the threshold T_{TRANSMIT} . If so, the counter $C(i,j)$ is
 5 incremented by one in function block 704, and then a test is made in decision block 705 to determine if the value of counter $C(i,j)$ is less than T_{RESET} . If not, the counter $C(i,j)$ is reset to zero in function block 706 before the process loops back to decision block 701; otherwise, the process loops back to decision block 701 directly. If all slaves have been checked, then the slave
 10 with the lowest counter value is chosen as the current slave and the packet size is set to one in function block 707.

After look-ahead of the transmission frequency f_j , the master selects a slave for transmissions as per the flowchart embodied in Figure 7A. If the link state history counter for this slave "i" at the expected transmission frequency f_j
 15 indicates that a transmission attempt can be made, (i.e., value in $C(i,j)$ is less than or equal to T_{TRANSMIT}) the master proceeds to transmit to the slave at an appropriate packet size. But, if the counter value forbids transmission, the master increments the counter $C(i,j)$ by one. If after the increment operation the counter value becomes greater than T_{RESET} , it is reset to zero. Then the
 20 master tries to choose another active slave if any for transmission. In this case, the loss of service by the intended slave and gain of service by another slave can be recorded to address the issue of service fairness as per any suitable policy. In the event that no other slave can be found, the master chooses the slave whose link state history counter has the lowest value and decides on a
 25 packet size of one.

Packet Size Selection

Continuing the description of the flow chart, the process next goes to Figures 7B and 7C, where the packet size determination is done using the link state history counters. In these figures, the frequencies on which the slave has to respond if the master transmitted one, three or five size packets are called, respectively, f_{k_2} , f_{k_4} and f_{k_6} . Now, a test is made in decision block 711 to determine if the value of counter $C(i,k_6)$ is greater than the threshold T_{TRANSMIT} . If not, the packet size is set to five in function block 712; otherwise, the counter $C(i,k_6)$ is incremented by one in function block 713. A test is made in decision block 714 to determine if the value of counter $C(i,k_6)$ exceeds the threshold T_{RESET} . If so, counter $C(i,k_6)$ is reset to zero in function block 715; otherwise, no change is made to the counter. A further test is made in decision block 716 to determine if the counter $C(i,k_4)$ is greater than the threshold T_{TRANSMIT} . If not, the packet size is chosen as three in function block 717; otherwise, the process goes to decision block 721 in Figure 7C where a test is made to determine if counter $C(i,k_4)$ is greater than the threshold T_{RESET} . If so, the counter is set to zero in function block 722; otherwise, no change is made to the counter. A further test is made in decision block 723 to determine if the value of counter $C(i,k_2)$ is greater than the threshold T_{TRANSMIT} . If not, the packet size is chosen as one in function block 724; otherwise, the counter $C(i,k_2)$ is incremented by one in function block 725. A test is next made in decision block 726 to determine if the value of counter $C(i,k_2)$ exceeds the threshold T_{RESET} . If so, the counter is reset to zero before the process loops back to decision block 701 in Figure 7A; otherwise the process loops back to decision block 701 without making any change to the counter.

As indicated in the flowcharts of Figures 7B and 7C, once the slave for transmission has been determined, the master considers the sixth time slot

frequency f_{k_6} and finds its transmission worthiness for the slave by referring to the corresponding link state history counter $C(i, k_6)$. If the frequency is found good, the master chooses to transmit a five size packet to the slave; otherwise, the master increments the counter $C(i, k_6)$ by one and if after the

5 increment operation the counter value becomes greater than T_{RESET} , it resets the counter to zero, and then proceeds to check the fourth time slot frequency f_{k_4} for transmitting a three size packet. If the fourth time slot frequency is good, a three size packet is chosen for transmission. In the event that even that frequency is bad, $C(i, k_4)$ is incremented by one and reset to zero if its value

10 is greater than T_{RESET} after increment operation. Then, the master considers the second time slot frequency f_{k_2} , and if found good, chooses a packet of size one to transmit. If however f_{k_2} is also bad, the master increments the counter $C(i, k_2)$ by one and if after the increment operation the counter value becomes greater than T_{RESET} , it resets the counter to zero, and then proceeds to choose

15 another slave for transmission. It may be noted that when a multi-slot packet is chosen, some frequencies corresponding to intermediate time slots are masked with reference to any interference. Thus, if a five-slot packet is chosen, the transmission is independent of interference in frequencies corresponding to the second through fifth slots. Similarly, with a three-slot packet, interference

20 for second and third slot frequencies does not affect the packet transmission.

Goodness Counter Based Link State History

Another embodiment of the invention is an alternative method of maintaining short term link state history which involves transfer of link state information between the different slaves and the master. Every active slave is

25 required to monitor the transmissions from the master in expectation of a packet being sent to it. If a packet is not meant for a slave, then it can go to

sleep state until the next packet transmission from the master. In this alternative method of link state history maintenance, slave “ i ” maintains a “goodness” counter $GC(i,j)$ for each frequency f_j . The counter $GC(i,j)$ is incremented every time the slave “ i ” successfully receives a valid packet header from a master transmission on frequency f_j . However, if an error in receiving packet header is detected the goodness counters are unaltered. An error in receiving a master’s packet header can occur if there is interference in the master’s transmission frequency or if the even numbered time slot is being used by another slave to transmit a three or five sized packet. The counter is also incremented when a packet transmitted by the slave “ i ” on frequency f_j is successfully acknowledged by the master.

In this method, in response to periodic requests by the master, the active slaves in the pico-cell send the values of the goodness counters to the master. After successfully transmitting the goodness counter values, each slave resets its goodness counters to zero for link state monitoring during the next period of communication. The values of the counters received from all the slaves are compiled by the master to form a link state history table of goodness counters as shown in Table 2 for use during the next scheduling period. In Table 2, $GC(i,j)$ is the goodness counter for the link state of frequency f_j for slave i .

Table 1. Link State History Table of Goodness Counters

	f_1	f_2	f_3		f_{78}	f_{79}
SLAVE 1	GC(1,1)	GC(1,2)	GC(1,3)	...	GC(1,78)	GC(1,79)
SLAVE 2	GC(2,1)	GC(2,2)	GC(2,3)	...	GC(2,78)	GC(2,79)
SLAVE 3	GC(3,1)	GC(3,2)	GC(3,3)	...	GC(3,78)	GC(3,79)
5 SLAVE 4	GC(4,1)	GC(4,2)	GC(4,3)	...	GC(4,78)	GC(4,79)
SLAVE 5	GC(5,1)	GC(5,2)	GC(5,3)	...	GC(5,78)	GC(5,79)
SLAVE 6	GC(6,1)	GC(6,2)	GC(6,3)	...	GC(6,78)	GC(6,79)
SLAVE 7	GC(7,1)	GC(7,2)	GC(7,3)	...	GC(7,78)	GC(7,79)

As the number of bits in the counter is to be limited to minimize the overhead in transmitting the information to the master, the counters $GC(i,j)$ are allowed to count up to the maximum value and stay there until reset. The value of any counter $GC(i,j)$ received by the master indicates the relative goodness of the link between the master and slave “ i ” on the frequency f_j ; the higher the count value, the better the link. In this method, as the slaves listen to all the master transmissions and record the successful transmissions, the monitoring of interference on different frequencies occurs more frequently, and therefore results in better characterization of the link states.

Slave Selection Based on Goodness Counters

A flowchart for selection of a slave for packet transmission is shown in Figure 8A. A test is made in decision block 801 to determine if all the active slaves have been checked for scheduling. If not, the next slave is picked as the current slave “ i ” in function block 802. A test is then made in decision block 803 to determine if the value of counter $GC(i,j)$ is less than the threshold T_{GOOD} . If so, the process loops back to decision block 801; otherwise, the

process goes on to the flowchart of Figure 8B. If all the active slaves have been checked for scheduling, as determined in decision block 801, the slave with the highest goodness counter value is chosen as the current slave in function block 804. The packet size is set to one.

- 5 In the process shown in Figure 8A, the master picks up the next slave for transmission from among the currently active slaves. The first slave for which the value of goodness counter $GC(i,j)$ for the frequency of transmission f_j is greater than or equal to a minimum goodness threshold value T_{GOOD} is chosen and a suitable packet size decided next. If however none of the slaves' goodness counter value is greater than or equal to T_{GOOD} , the master chooses the slave with the highest value of the goodness counter and decides on a packet size of one.
- 10

Packet Size Selection Based on Goodness Counters

- As shown in Figure 8B, the master first checks in decision block 811 if the slave's response frequency f_{k_6} is good as indicated by the value of the counter $GC(i,k_6)$. If the value of the goodness counter $GC(i,k_6)$ is greater than or equal to T_{GOOD} , then the master takes the decision of sending a five-slot packet in function block 812. If however $GC(i,k_6)$ value forbids the transmission of a five-slot packet, then the master checks in decision block 813 for the value of the goodness counter $GC(i,k_4)$. If $GC(i,k_4)$ value is greater than or equal to T_{GOOD} , then the master takes the decision of sending a three-slot packet in function block 814. However, if $GC(i,k_4)$ value also forbids transmission, the master checks in decision block 815 for the value of the goodness counter $GC(i,k_2)$ for transmitting a one-slot packet and, if possible, chooses a one size packet in function block 816. In the event that even $GC(i,k_2)$ value is also below T_{GOOD} , the master does not transmit to the particular slave, and proceeds to choose another slave for transmission by
- 15
- 20
- 25

looping back to decision block 801 of Figure 8A.

Second Level Look-Ahead

Generally, a single time slot packet transmission occurs within the first time period T_1 , out of the total time slot period of T_s , as shown in Figure 9.

5 Thus, time period of $(T_s - T_1)$ is available for any processing after completely receiving a packet. However, the packet size being transmitted is known by the receiving unit as soon as the packet header is received during the beginning of the time slot. Therefore, the processor in the master can perform the processing required to decide the choice of slave and packet size to be
10 transmitted soon after receiving the header of a packet sent by a slave. This provides a time period of at least about one time slot for deciding about the next slave and slot size. However, a second level of look-ahead can be performed as described below to accommodate slower speed processors.

As shown in Figure 10, after a master transmits a packet in time slot
15 TS0, the next possible time slots for its transmission are TS1, TS3 or TS5 depending on whether the addressed slave transmits a one, three, or five slot packet, respectively. Therefore, soon after the master's transmission in slot TS0, a look-ahead for frequencies of transmission during TS1, TS3, and TS5 can be performed and the corresponding slave(s) and packet sizes can be
20 determined in advance. This method provides at least about a time period of two time- slots for the required processing steps.

While the invention has been described in terms of preferred
embodiments, those skilled in the art will recognize that the invention can be
practiced with modification within the spirit and scope of the appended
25 claims.